

Tema 1. Introducción a las bases de datos

Indice

1. Los sistemas de almacenamiento de la información.....	1
1.1. Ficheros.....	1
1.1.1. Tipos de ficheros.....	2
1.2. Ficheros texto.....	3
1.3. Ficheros binarios.....	4
2. Bases de Datos.....	5
2.1. Conceptos.....	6
2.2. Estructura de una base de datos.....	7
2.3. Usos de las bases de datos.....	8
2.4. Evolución histórica.....	9
2.5. Tipos de bases de datos.....	10
3. Los Sistemas Gestores de Bases de Datos.....	11
3.1. Concepto de Sistema Gestor de Base de Datos.....	11
3.2. Funciones de un SGBD.....	11
3.3. El lenguaje SQL.....	12
3.4. Tipos de SGBD.....	13

1. Los sistemas de almacenamiento de la información

1.1. Ficheros

En un ordenador se almacenan muchos tipos de información desde datos administrativos o bancarios hasta imágenes, fotos, música etc. Esta información se almacena en los dispositivos de almacenamiento: disco duro, pen drive, dvd etc. La forma de organizar la información en estos dispositivos es mediante ficheros o archivos, por tanto podemos definir los archivos como estructuras de información que crean los sistemas operativos para almacenar los datos. Generalmente estos archivos tienen un nombre y una extensión

que determina el tipo de información que contienen.

1.1.1. Tipos de ficheros

El formato y el tipo de ficheros determina la forma de interpretar la información que contiene ya que lo único que se almacena en un fichero es una ristra de bits (ceros y unos). Por ejemplo para almacenar una imagen se puede usar un fichero binario bmp. En este formato se almacena un vector de datos con los colores que cada uno de los píxeles de la imagen, es decir que la información se almacena en un determinado formato. Para poder abrir correctamente el archivo se debe utilizar la aplicación adecuada capaz de reconocer formato del fichero.

Tradicionalmente los ficheros se pueden clasificar según diferentes criterios:

- Por **contenido** se clasifican en **ficheros texto** donde los bits pueden ser traducidos por el sistema operativo y en **ficheros binarios** en los cuales los bits se estructuran en formatos más complejos que requieren aplicaciones específicas para ser interpretados, por ejemplo videos, imágenes, sonidos etc.
- La **organización** indica la forma en que se accede al fichero. Según la organización los ficheros se clasifican en **secuenciales** en los cuales los datos se leen unos detrás de otros, es decir que para leer el dato n previamente se han leído los $n-1$ anteriores. En los ficheros **directos** se puede acceder a un dato en concreto sin necesidad de acceder a todos los anteriores. Finalmente los ficheros indexados utilizan un índice, es decir una estructura de datos que permite acceder al contenido del fichero de manera similar a como se hace con el índice de un libro (podemos acceder directamente al apartado o tema de un libro mediante el número de página).
- La **utilidad** de un fichero indica que uso se va a hacer de él por ejemplo en una organización los clientes se almacenan en un fichero principal llamado **maestro**, las variaciones (altas, bajas y modificaciones) se registran en ficheros de **movimientos** y cuando los datos nos son necesarios en el proceso diario se guardan en ficheros **históricos**.

Actualmente los ficheros se tratan desde dos puntos de vista:

- El contenido: texto y binarios
- El tipo: imágenes, audio, ejecutables etc.

1.2. Ficheros texto.

Los ficheros de texto también suelen llamarse ficheros planos o ficheros ascii (American Standard Code for Information Interchange). Es un estándar que asigna un valor numérico a cada carácter. Se utiliza 1 byte (8 bits) para representar cada carácter.

Con cada byte se pueden generar $2^8 = 256$ caracteres distintos, numerados del 0 al 255.

A tener en cuenta:

- Los 32 primeros caracteres son no imprimibles y se utilizan tradicionalmente para el control de las transmisiones.
- La distancia entre mayúsculas y minúsculas es de 32 caracteres. Por ejemplo la **A** es el 65 y la **a** es el 97.
- El valor ascii de los caracteres numéricos es el resultado de sumar 48 a su valor. Por ejemplo el valor ascii del carácter 6 es $6+48=54$.

Ctrl	Dec	Hex	Car.	Código	Dec	Hex	Car.	Dec	Hex	Car.	Dec	Hex	Car.
^@	0	00		NUL	32	20	!	64	40	@	96	60	'
^A	1	01		SOH	33	21	!	65	41	A	97	61	a
^B	2	02		STX	34	22	"	66	42	B	98	62	b
^C	3	03		ETX	35	23	#	67	43	C	99	63	c
^D	4	04		EOT	36	24	\$	68	44	D	100	64	d
^E	5	05		ENQ	37	25	%	69	45	E	101	65	e
^F	6	06		ACK	38	26	&	70	46	F	102	66	f
^G	7	07		BEL	39	27	'	71	47	G	103	67	g
^H	8	08		BS	40	28	(72	48	H	104	68	h
^I	9	09		HT	41	29)	73	49	I	105	69	i
^J	10	0A		LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	+	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	0	80	50	P	112	70	p
^Q	17	11		DC1	49	31	1	81	51	Q	113	71	q
^R	18	12		DC2	50	32	2	82	52	R	114	72	r
^S	19	13		DC3	51	33	3	83	53	S	115	73	s
^T	20	14		DC4	52	34	4	84	54	T	116	74	t
^U	21	15		NAK	53	35	5	85	55	U	117	75	u
^V	22	16		SYN	54	36	6	86	56	V	118	76	v
^W	23	17		ETB	55	37	7	87	57	W	119	77	w
^X	24	18		CAN	56	38	8	88	58	X	120	78	x
^Y	25	19		EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z
^[27	1B		ESC	59	3B	;	91	5B	[123	7B	{
^\	28	1C		FS	60	3C	<	92	5C	\	124	7C	
^]	29	1D		GS	61	3D	=	93	5D]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
^-	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	␣*

* El código ASCII 127 tiene el código DEL. En MS-DOS, éste código tiene las mismas funciones que ASCII 8 (BS). El código DEL se puede generar con las teclas CTRL + RETROCESO.

Figura 1.1: Tabla ascii

Aunque los ficheros de texto no necesitan un formato para ser interpretados (se pueden abrir con cualquier editor de texto) suelen tener diferentes extensiones que indican el tipo de información que contienen. Por ejemplo:

- **Ficheros de configuración:** Su contenido permite configurar el sistema operativo o aplicaciones. En windows los ficheros de configuración llevan la extensión .ini mientras que en linux la extensión es .conf.
- **Ficheros de código fuente:** su contenido son el código de los programas, por ejemplo .java .c
- **Ficheros de páginas web:** Las páginas web son ficheros con hipertexto¹, por ejemplo .html, .css .xml
- **Formatos enriquecidos:** Son textos que contienen códigos de control para ofrecer una visión más elegante, por ejemplo .rft, .ps



XML es un lenguaje estándar para el intercambio de datos entre aplicaciones informáticas. La mayoría de aplicaciones permiten exportar e importar datos en formato XML. Actualmente se están desarrollando las bases de datos nativas XML cuyo objetivo es el almacenamiento de documentos texto con código en XML. Existen lenguajes de consulta que actúan directamente sobre documentos XML.

1.3. Ficheros binarios.

Los ficheros binarios son todos aquellos que no son texto y requieren un formato para ser interpretados. Algunos ejemplos de ficheros binarios:

- Imágenes: .jpg, .bmp, .png, .gif
- Vídeo: .mpg, .avi .mov
- Comprimidos o empaquetados: .zip, .gz, .tar
- Ejecutables o compilados: .exe, .com
- Procesadores de texto: .doc, .odt

Generalmente los ficheros que componen una base de datos son de tipo binario, ya que la

1 El hipertexto es una forma de escritura no secuencial, con bifurcaciones, que permite al lector elegir la secuencia a seguir y que es presentado en una pantalla interactiva para facilitar la navegación

información almacenada tiene una estructura lógica y organizada difícil de expresar mediante ficheros texto. La información de una base de datos se puede guardar en un solo fichero por ejemplo en acces (.mdb) o en varios ficheros como en MySql u Oracle.

2. Bases de Datos.

Una **Base de datos** es una colección de información perteneciente a un mismo contexto que se almacena de forma organizada en ficheros.

Una base de datos relacional está organizada mediante **tablas** que almacenan información concerniente a algún objeto o suceso. Estas tablas se relacionan formando vínculos o **relaciones** entre ellas, que ayudan a mantener la información ente los diversos objetos de forma ordenada y coherente (sin contradicciones). Cada una de estas tablas es una estructura parecida a una hoja de cálculo ya que están dispuestas en filas y columnas. Cada fila almacena un **registro** con tantos **campos** como columnas tenga la tabla.

	Expediente	Nombre	Apellidos	Grupo	FechaNacimiento
1		Veronica	Romero Milheirico	1	28/04/89
2		Rubén	Durán Milheirico	1	28/04/89
3		Manuel	Moreno Martín	1	13/06/90
4		Juan Diego	González Pulido	1	23/08/90
5		Jesús	Naranjo Charro	1	20/02/90
6		Manuel	Álvarez Menor	1	04/02/90
7		Cristian	Aguado Caro	1	14/11/89
8		Alejandro	Antonio González	1	15/03/90
9		María	Caballo Santos	1	18/01/90
10		Gerardo	Correa Morán	1	04/09/91
11		Sheila	Cosme Almeida	1	29/08/92
12		Rubén	Crespo Bonilla	1	04/08/90
13		María Dolore	Durán Rasero	1	22/08/89
14		Sheila	Eduardo Sánchez	1	04/07/87
15		Tamara	Flores Hernández	1	08/02/88
16		José Daniel	García López	1	10/04/89
17		Lorena	García Ortiz	1	12/07/88
18		Clara	Gil González	1	01/04/89
19		Claudia	Guerrero Hernández	1	04/07/89
20		Daniel	Infantes Rubio	1	20/05/90
21		Sonia	López González	1	24/03/90
22		Noelia	Marcos Gordillo	1	13/06/89
23		Sara	Marín Gallego	1	17/05/90
24		Gemma Mar	Martín Pastor	1	17/05/90
25		Alicia María	Moreno Barbosa	1	29/12/91
26		Marta	Muniz Alba	2	12/01/94
27		Rubén	Nicolas Pablos	2	17/10/94
28		Jessica	Nogales Gordillo	2	08/09/95
29		José Antonio	Pardo Montes	2	23/08/94

Figura 1.2: Tabla de alumnos

En la tabla Alumnos de la imagen, cada fila o registro son los datos de un alumno y cada campo representa una parte de la información de cada alumno por ejemplo los apellidos o la fecha de nacimiento.

2.1. Conceptos.

Algunos de los conceptos que nos vamos a encontrar en las bases de datos son:

- **Dato:** El dato es un trozo de información concreta sobre algún concepto o suceso. Por ejemplo *García* es el apellido de una persona. Los datos se caracterizan por pertenecer a un tipo.
- **Tipo de Dato:** El tipo indica la naturaleza del campo. En una primera clasificación podemos tener *datos numéricos*, que son aquellos que permiten realizar cálculos aritméticos (sumas, multiplicaciones ...) y los *datos alfanuméricos* que contiene caracteres alfabéticos y dígitos numéricos.
- **Campo:** Un campo es un identificador para toda una familia de datos. Cada campo pertenece a un tipo de datos. Por ejemplo apellidos es un campo de tipo cadena. Al campo también se le llama *columna*.
- **Registro:** Es una recolección de datos referente a un mismo concepto o suceso. Por ejemplo los datos del alumno Juan Diego

4	Juan Diego	González Pulido	1	23/08/90
---	------------	-----------------	---	----------

Figura 1.3: registro de la tabla alumnos

A los registros también se les llama **tuplas** o **filas**.

- **Campo Clave:** Es un campo especial que identifica de forma única a cada registro, por ejemplo el NIF de cada persona es único, por tanto es un campo clave.
- **Tabla:** Es un conjunto de registros bajo el mismo nombre que representa el conjunto de todos ellos. Por ejemplo la tabla alumnos de la base de datos del instituto.
- **Consulta:** Es una instrucción para hacer peticiones a una base de datos. Puede ser una búsqueda simple de un registro específico, por ejemplo “*el alumno con expediente 4*” o bien consultas más complejas para seleccionar los registros que cumplen unos criterios, por ejemplo “*los alumnos mayores de edad que cursan bases de datos y hablan chino*”.
- **Índice:** Es una estructura que almacena algunos campos de una tabla, organizándolos para facilitar la ordenación y la consulta de los registros de una tabla. El índice tiene un funcionamiento similar al índice de un libro, guardando

parejas de elementos: el elemento que se desea indexar y su posición en la base de datos. Para buscar un elemento que esté indexado solo hay que buscar en el índice dicho elemento para que una vez encontrado devuelva el registro correspondiente a la posición marcada por el índice. Por ejemplo podemos tener un índice con el campo fecha de nacimiento que nos permita buscar más rápidamente que alumnos cumplen los años un determinado día.

- **Vista:** Es una transformación que se hace a una o más tablas para obtener una nueva tabla virtual, es decir no existe físicamente, solo se almacena su definición, pero puede ser tratada como una tabla física por la aplicación de base de datos. Por ejemplo puedo definir una vista alumnos_bd que solo tenga los alumnos que cursan el módulo de base de datos.
- **Informe:** Es un listado ordenado de los campos y seleccionado en un formato fácil de leer. Generalmente se usan como peticiones expresas de un tipo de información por parte de un usuario, por ejemplo las notas de los módulos de los alumnos de primero de DAW.
- **Guiones o scripts:** Son un conjunto de instrucciones (líneas de código) que ejecutadas de forma ordenada realizan operaciones avanzadas de los datos almacenados en la base de datos.
- **Procedimientos:** Son un tipo especial de script que está almacenado en la base de datos y que forma parte de su esquema.

2.2. Estructura de una base de datos.

Una base de datos almacena los datos a través de un esquema. Un **esquema** es la definición de la estructura donde se almacenan los datos, contiene todo lo necesario para organizar la información mediante tablas, registros y campos. También contiene otros objetos necesarios para el tratamiento de los datos como los índices, vistas, procedimientos etc. Al esquema también se le suele llamar *metainformación*, es decir, información sobre los *metadatos* (datos que definen la estructura de la base de datos).

Algunos gestores de bases de datos como Oracle o MySQL almacenan el esquema de base de datos en tablas. Por ejemplo queremos ver las tablas y número de filas de cada tabla de la base de datos liga.

table_schema	table_name	table_rows
liga	equipo	8
liga	jugador	22
liga	partido	16

Figura 1.4: metadatos de la base de datos liga

2.3. Usos de las bases de datos.

Las bases de datos se utilizan en todo tipo de aplicaciones, por ejemplo:

- Bases de datos administrativas: en cualquier empresa para registrar clientes, pedidos, facturas etc.
- Bases de datos contables: para gestionar balances, pagos, perdidas y ganancias etc.
- Bases de datos para motores de búsqueda, por ejemplo Google tiene una base de datos gigantesca para almacenar información de todos los documentos de Internet.
- Bases de datos científicas: recolección de datos meteorológicos, geológicos etc.
- Bibliotecas
- Censos
- Virus
- y muchos más



La WDCC (World Data Climate Center), centro mundial para datos del clima, es la base de datos más grande del mundo. Almacena alrededor de 6 petabytes (6144 Terabytes) de información sobre clima, predicciones y simulaciones

2.4. Evolución histórica.

Primera generación

- Partieron de los sistemas basados en archivos, empezando a desarrollarse para el proyecto lunar Apollo en la década de los 60's desarrollándose GUAM (Generalized Update Access Method)
- A mediados de los 60 apareció IDS (Integrated Data Store) de General Electric y IMS (Integrated Management System) de IBM.
- En 1967 se creó la organización DBTG (Data Base Task Group) para la especificación de un estándar dando origen a CODASYL o DBTG.
- Problemas de programación compleja para acciones simples, mínima independencia de los datos y no existía una base teórica aceptada.

Segunda generación

- Codd de IBM definió el modelo de datos relacional abriéndose el campo de las BD comerciales.
- Proyecto System R de IBM que condujo a desarrollar SQL e implementación de productos como DB2, SQL/DB y Oracle.
- Todo ello dio paso a los SGBD relacionales.
- Sistemas orientados a los datos. A finales de los 60's: los datos se organizan y mantienen en un conjunto estructurado que no está diseñado para una aplicación concreta; satisface las necesidades de información de toda la organización.

Tercera generación

- Creciente complejidad de los datos y las aplicaciones que los tratan.
- OODBMS: bases de datos orientadas a objetos.
- ORDBMS: bases de datos objeto-relacionales.
- Amplían la expresividad pero se alejan del modelo relacional original.

2.5. Tipos de bases de datos.

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se este manejando, o la utilidad de la misma.

Según la variabilidad de los datos almacenados:

- **Bases de datos estáticas:** Estas son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.
- **Bases de datos dinámicas:** Estas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado e inserción de datos, además de las operaciones fundamentales de consulta.

Además de la clasificación por la función de las bases de datos, estas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos(algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores.

Algunos de los modelos utilizados en las bases de datos son:

- **Bases De Datos Jerárquicas:** Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica (en forma de árbol).

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

- **Bases de datos de Red:** En este modelo se permite que un nodo tenga varios padres . Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos.
- **Bases de datos Transacciones:** Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial.

- **Bases de datos Relacionales:** Es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Cada relación es una tabla que está compuesta por registros (las filas de una tabla), y campos (las columnas de una tabla).
- **Bases de datos Multidimensionales:** Son bases de datos ideadas para desarrollar aplicaciones muy concretas.
- **Bases de datos Orientadas a Objetos:** Este modelo, bastante reciente, y propio de los modelos informáticos enfocado a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).
- **Bases de datos Documentales:** Permiten la indexación de textos y línea para generar búsquedas más potentes.
- **Bases de datos Deductivas:** Permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática.

3. Los Sistemas Gestores de Bases de Datos.

3.1. Concepto de Sistema Gestor de Base de Datos.

Se define un Sistema Gestor de Base de datos, en adelante SGBD, como un conjunto de herramientas que facilitan la consulta, uso y actualización de una base de datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Algunos ejemplos de SGBD son Oracle, MySQL, Microsoft SQL Server, Access, FoxPro, PostgreSQL, etc.

3.2. Funciones de un SGBD.

Los SGBD del mercado cumplen con casi todas las siguientes funciones:

1. Permiten a los usuarios almacenar datos, acceder a ellos y actualizarlos de forma sencilla y con un gran rendimiento, ocultando la complejidad y las características

físicas de los dispositivos de almacenamiento.

2. Garantizan la integridad de los datos, es decir, no permiten operaciones que dejen cierto conjunto de datos incompletos o incorrectos.
3. Integran, junto con el sistema operativo, un sistema de seguridad que garantiza el acceso a la información exclusivamente a aquellos usuarios que dispongan de autorización.
4. Proporcionan un diccionario de metadatos, que contiene el esquema de la base de datos.
5. Permiten el uso de transacciones, garantizan que todas las operaciones de la transacción se realicen correctamente, y en caso de alguna incidencia, deshacen los cambios sin ningún tipo de complicación adicional.
6. Ofrecen estadísticas sobre el uso del gestor, registrando operaciones efectuadas, consultas solicitadas, operaciones fallidas y cualquier tipo de incidencia.
7. Permiten la concurrencia, es decir, varios usuarios trabajando sobre un mismo conjunto de datos.
8. Independizan los datos de la aplicación o usuario que esté utilizándolos, haciendo más fácil su migración a otras plataformas.
9. Ofrecen conectividad con el exterior. De esta manera, se puede replicar y distribuir bases de datos. Además, todos los SGBD incorporan herramientas estándar de conectividad.
10. Incorporan herramientas para la copia de seguridad y restauración de la información en caso de desastre. anterior e el tiempo. Además, deben ofrecer sencillas herramientas para la importación y exportación automática de la información.

3.3. El lenguaje SQL.

El lenguaje SQL (Structured Query Language) consiste en un lenguaje muy sencillo mediante el cuál el usuario realiza preguntas al servidor. Este lenguaje está estandarizado por la ISO5, es decir, todas las bases de datos que soporten SQL deben tener la misma sintaxis a la hora de aplicar el lenguaje.

Se divide en 4 sublenguajes

- **Lenguaje DML:** o lenguaje de manipulación de datos (Data Manipulation Language). Este lenguaje permite con 4 sentencias sencillas seleccionar determinados datos (SELECT), insertar datos (INSERT), modificarlos (UPDATE) o incluso borrarlos (DELETE). En temas posteriores se desarrollará con profundidad la sintaxis de cada una de estas sentencias.
- **Lenguaje DDL:** o lenguaje de definición de datos (Data Definition Language). Este lenguaje permite crear toda la estructura de una base de datos (desde tablas hasta usuarios). Sus cláusulas son del tipo DROP (Eliminar objetos) y CREATE (Crear objetos).

3.4. Tipos de SGBD.

Los SGBD se pueden clasificar de muchas formas, por ejemplo, según las bases de datos que gestionan, según traten bases de datos relacionales, de datos orientadas a objetos, etc.

Podemos clasificar los SGBD según su capacidad y potencia del propio gestor en:

- **Los Gestores de Bases de Datos ofimáticas** son aquellos que manipulan bases de datos pequeñas (ofimáticas) orientadas a almacenar datos domésticos o de pequeñas empresas. Un ejemplo de un SGBD ofimático es Microsoft Access, que posee tanto una interfaz de usuario muy sencilla como un potente lenguaje de programación (VBA=Visual Basic for Applications) para ofrecer a usuarios avanzados muchas posibilidades de gestión mucho más específicas.
- **Los Gestores de bases de datos Corporativas** son aquellas que tienen la capacidad de gestionar bases de datos enormes, de grandes o medianas empresas con una carga de datos y transacciones que requieren un servidor de grandes dimensiones (generalmente un Servidor Unix, o un Windows 200X Server con altas prestaciones) Estos gestores son capaces de manipular grandes cantidades de datos de forma muy rápida y eficiente para poder resolver la demanda de muchos (cientos de usuarios). Un ejemplo típico de servidor de base de datos Corporativas es el anteriormente comentado Oracle, actualmente, junto con DB2, el servidor de base de datos más potente del mercado (también el más caro).

Precisamente, ese coste tan alto es que ha desencadenado que se haya recurrido a una solución intermedia entre gestores de base de datos ofimáticas y corporativas. Entre estas soluciones intermedias se encuentra MySQL, un gestor de base de datos que, además de ser gratuito y sencillo, es capaz de manipular gran cantidad de datos cumpliendo prácticamente todos los estándares de la arquitectura ANSI SPARC². Aunque implementa SQL, no tiene un lenguaje de

² La arquitectura ANSI-SPARC, donde ANSI-SPARC significa **American National Standards Institute, Standards Planning And Requirements Committee**, es un estándar de diseño abstracto para un sistema de gestión de bases de datos (DBMS), propuesto por

programación propio como SQL Server u Oracle pero a cambio se integra fácilmente en las típicas soluciones XAMPP, que son paquetes que incluyen, además de MySQL, una versión del servidor Web Apache y varios lenguajes de script (php, perl...) que dotan a MySQL de potentes herramientas para acceso y publicación de los datos.

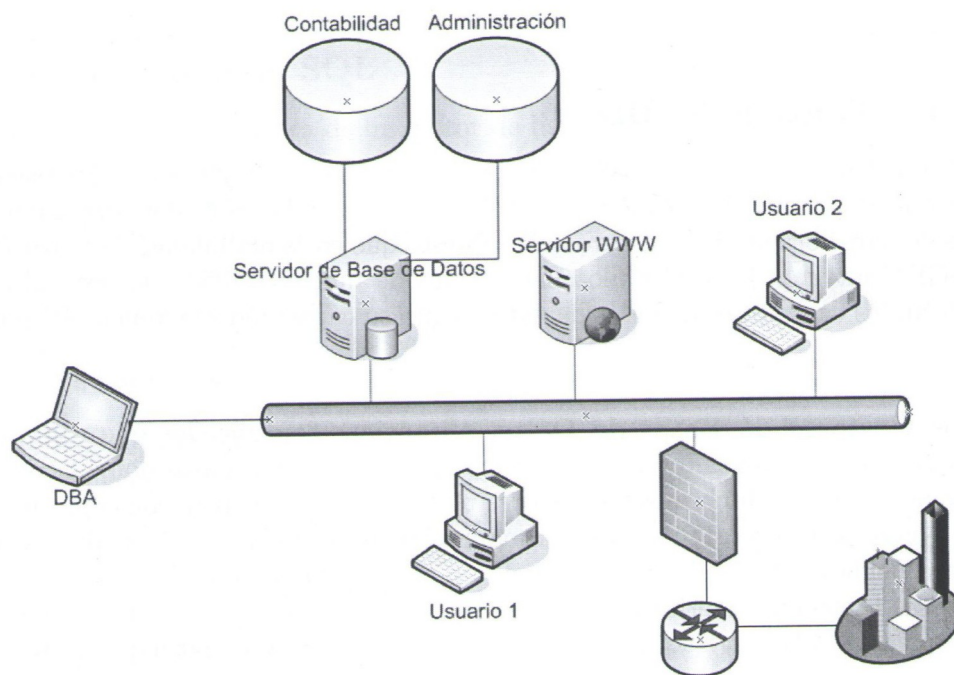


Figura 1.5: Esquema típico de organización de un SGBD corporativo

Bibliografía.

- LOPEZ MONTALBÁN, I. DE CASTRO VÁZQUEZ, M. OSPINO RIVAS, J. (2014). Bases de Datos. Madrid:Garceta.
- Overblog
<http://basededatos.over-blog.net/article-tipos-de-bases-de-datos-68319538.html>